Case-Centered Learning of Scientific Computing

Jarmo Rantakokko

Scientific Computing, Department of Information Technology
Uppsala University, Uppsala, Sweden

August 31, 2006

Abstract

Traditionally courses in scientific computing teach a large number of methods and algorithms for different mathematical problems. The algorithms are applied on simplified problems and not on real applications. The result is that the students can't see the main thread, they focus only on the details of the methods and don't see the entirety. The students can not put what they learn into perspective and their motivation to study becomes diminished. In this paper we suggest a case-centered approach for learning scientific computing. We use a real-life case, weather prediction, as a starting point for learning. The case is analyzed and discussed in class. To follow up the discussions the students are assigned learning tasks in scientific computing defined from the case analysis. The real-life application connects the different topics in scientific computing together and motivates the students. The response from the students has been positive and the case has increased their understanding of what scientific computing is and what it can be used for.

1 Introduction

In todays computerized society scientific computing is a science that affects everyone. We see every-day the results of scientific computing but still we are not aware of it. Scientific computing is about using numerical methods on computers to solve complex mathematical problems, problems that are too big or complex to be solved by analytical means. These problems are common in engineering and natural sciences, e.g., in physics, chemistry, biology and medicine. Scientific computing is connected to applications in these other sciences, it provides the methods and tools to produce results. This makes scientific computing a *hidden science*, it is the science of the application that is visible while scientific computing is used in the background. As an example we can look at weather prediction. Here, we immediately associate to meteorology while in the background lots of methods from scientific computing are used. The starting point for a weather prediction is a description of the dynamics in the atmosphere with mathematical equations, i.e., partial differential equations. These equations can only be solved with numerical methods developed in scientific computing. The results of the computations are then interpreted by a meteorologist who presents the final weather map. The numerical methods in the process are not visible to the general public, they are *hidden*.

In the teaching of scientific computing the conditions are the opposite. Here, the students learn about different methods and algorithms in isolation from real applications. The methods are applied to simplified problems which often can be solved by analytical means as well. The result is that the students can not put what they learn into perspective, they don't understand why they should learn the subject and the motivation to study becomes diminished. This problem is especially evident for computer science majors who don't see the demanding applications in chemistry, physics, biology, or medicine. There is a need for a change of pedagogy from traditional and artificial problem solving sessions on the black board to more reality based case sessions.

2 Case pedagogy

Case pedagogy has traditionally been used in law, business, teacher training, and social science classes. In these disciplines the cases are open and there is not a correct answer or solution. Often it is not even clear what the problem is. Moreover, the cases are built on a story involving a number of persons or characters. These types of cases are very difficult to adopt in engineering and natural sciences where the problems are clear and they have (in most cases) an unique solution. However, the case pedagogy involves other issues that more easily can be adopted in engineering and natural sciences as well. The cases are reality based, real-life problems, and one of the main issues is to discuss and analyze the problems in class. The students become active, both in their preparation of the case discussion and during the case seminar where they can express their opinions and views.

Traditional lectures in scientific computing deal with artifical and simplified problems that are cleared up in order to be solved in class and the connection to real-life problems is not visible. A large number of methods and algorithms for different mathematical problems are taught to the students without any obvious coherence. Moreover, the simplified problems can more or less be solved mechanically without further analysis or discussions. The students become passive and non-critical in their learning and thinking process. Using case pedagogy, not necessarily in its pure form as described e.g. in [1, 2, 4] but still in using the spirit of it by analyzing real-life cases in class, can change and make the students more active in their learning process. Thus, activating the students in critical thinking, analysis and discussions. The discussions in class and within smaller discussion groups has showed to have positive effects on learning [3]. Moreover, the real-life cases connects methods and theory in the subject with problems they know from their everyday life, motivating learning and developing problem solving skills for realistic problems. Learning about real-life problems in project groups has been expressed by many teachers to be motivating and preparing the students for working life, see e.g. [5, 6, 7].

3 Case centered learning

Teaching scientific computing face several difficulties. The students are often unmotivated and don't see the main thread in the subject. As described above the problem is that when students see an application they don't see the use of scientific computing (which is a tool to get the results) and when they learn scientific computing they don't apply it to real applications. As a consequence, they can't put what they learn into a context and can't see the entirety.

Obviously there is a need to integrate numerical methods with real applications when teaching scientific computing. This is often done, but at a small extent, by briefly mentioning the areas of applications in an introductory lecture as a motivation for the whole course or in the beginning of a lecture as a motivation for a specific topic. Due to such a brief and limited introduction it is impossible to go into any details of the applications and the students can still not apply their knowledge to real cases.

By analyzing real applications more carefully it is possible to create a thread or a theme that ties the different methods and algorithms together giving an entirety and a context to the subject. At the same time it shows the usage of scientific computing applied to real cases. As an example we can look at weather prediction. Here, a number of different numerical methods from scientific computing are used in the different steps of the process:

1. *Data assimilation*
   Before we can do a prediction of future weather we must have data of an initial state of the weather, i.e., a starting point with data of the current weather situation. This data comes from weather observations, satellite pictures, balloon sondes, etc. The problem is that these weather observations contain errors and disturbances (measurement, apparatus, etc) and that they are irregularly distributed geographically (have insufficient coverage). These data are then used to produce an initial state that best adjusts to all observations and covers the whole domain of interest. To do this, numerical optimization methods from scientific computing are used (this process is called variational data assimilation).

2. *Solution of partial differential equations*
   The mathematical equations governing the dynamics of the atmosphere and controlling the weather are called partial differential equations. These equations can not be solved with traditional mathematical methods. They can only be solved with numerical methods on computers. A common approach in meteorology is to use *finite difference methods* to solve the equations. Finite difference methods is an important topic in scientific computing.
3. *Solution of linear systems of equations*
   The numerical solution of partial differential equations with finite difference methods can generate large and sparse systems of linear equations in the solution process. In principle it is possible to solve linear equations with mathematical methods (Gaussian elimination) but here the systems of equations become so large that it is impossible to use direct methods due to time and computer memory limitations. Instead the systems of equations are solved with iterative methods from scientific computing which are faster and requires less memory (exploiting the sparsity).
4. *Computations on parallel computers*
   The numerical solution of partial differential equations and linear systems of equations is still very demanding requiring very powerful computers, i.e., parallel computers to be used to get the results in time (while the computations are still a prediction). Efficient implementation and parallelization of numerical methods is also a topic in scientific computing.
5. *Interpolation of the solution*
   The numerical solution to partial differential equations is only represented in discrete points in the domain. These discrete points are typically in the range of 11 km apart in weather models. To get data of weather phenomena in between these solution points requires use of advanced interpolation methods, another topic in scientific computing.

As we can see with the case from weather prediction, using a case as a center for learning scientific computing different topics can be related together giving a context to the subject. The students can see the whole picture of scientific computing and not only the details of different methods.

4 Design of a project course

The idea of using case-centered learning has been implemented in the course *Scientific computing* for computer science majors at Uppsala University, spring semester 2006. The course contains two threads running simultaneously. The first thread is theoretical with traditional lectures, problem solving sessions, computer exercises and a written exam. Here, basic numerical methods in scientific computing are taught the traditional way. The second thread is project based, without a written exam, and the students are divided into groups of six persons. In this part the students learn more advanced numerical methods within the project groups and report to the others. Here, we have adopted the case-centered learning idea using weather prediction as a real-life case.

The project thread is divided into three phases. In the first phase all groups get the task to follow a weather prediction. They are asked to choose a city and specific day for the weather prediction. Then they follow the development of the prediction from five days ahead, two days ahead, one day ahead, to what the actual weather will be at their city. The task is to note any differences between the five day, two day, and one day prognosis compared to the real outcome, i.e., the actual weather. They should also *think* about what these differences can depend on and what sources of errors cause the deviations from the real weather. The task (case) is then discussed and analyzed in the class.

To follow up this discussion, the students are in the second phase asked to find out how a weather prediction is produced. They should describe the steps included in the process, especially what mathematical equations need to be solved and what numerical methods are used. The second phase is reported in writing but also discussed in a full class seminar.

As a result of phase two the project groups are assigned different study tasks, i.e., they are required to learn about a specific topic in scientific computing as described in Section 3. They are asked to explain the theory behind the algorithms and demonstrate how the methods work on a simplified model problem. The students must derive the model problem by themselves from the case. By letting them derive the model problem it not only clarifies the connection to real applications but also reflects their level of ambition and interest. As an introduction to the topic each group is given a short lecture. Then they are also assigned an expert in the topic who they can consult. The expert is an active researcher or a lecturer at the department. But most of the information they should find in the literature by their own. This last part of the project is also reported in writing, in a full detailed report, and in an oral presentation for the rest of the class.

5 Evaluation

The number of students taking the course were 40 and they were divided into 6 project groups. All students except one were approved on the project thread. The single student who was not approved quit the course of an unknown reason. The other students followed all deadlines, turned in all reports, and attended all seminars. The quality of the reports and the oral presentations were high, comparing with corresponding reports and presentations from previous years students. The seminar discussions were very fruitful. The students were active and complemented each other giving a comprehensive picture of the case. The students seemed to be motivated and worked sincerely with the tasks. These parameters are difficult to measure objectively, however, the students were also asked to fill in a course evaluation. The following questions related to project thread were asked:

1. What was your general opinion of the course?
2. Did the project thread feel relevant to the course?
3. Has the project given understanding of what scientific computing is?
4. Has the project given understanding of what scientific computing is useful for?
5. Has the project increased your motivation to learn scientific computing?
6. Has the project given training in searching of information, critical thinking and evaluation of knowledge, and presentation of results?

The students rated their answers from 1 to 5 where 1 is the lowest grade representing disagreement with the question and 5 is the highest grade representing full agreement with the question. The results of the evaluation are summarized in Figure 1 with the bars representing the average value for each question.

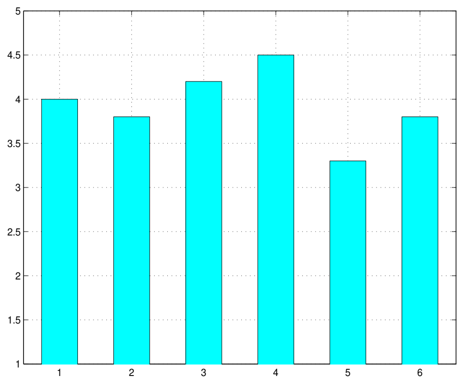

Figure 1: Average grading for the evaluation questions 1–6 above.

The results show a very good agreement for all question, and in particular for question 3 and 4. The case-centered learning approach has fulfilled our goals. It has in a high extent contributed to show what scientific computing is and what it can be used for. It has also motivated the students and activated them in critical thinking. The overall judgment of the course and the project is positive.

6 Conclusions

In this paper we have used a case-centered learning approach for scientific computing. We have used a real-life case, weather prediction, as a starting point for learning. The case has been analyzed and discussed in class. To follow up the seminars the students have been assigned learning tasks in scientific computing defined from the discussions.

The case-centered learning approach ties together different themes in scientific computing giving an entirety to the subject. Moreover, it connects the methods taught to real everyday applications motivating learning and developing problem solving skills for realistic problems. It also activates the students in their learning process and encourages to critical thinking.

The case seminars have been very fruitful, the students have been active giving a comprehensive description and analysis of the case. The quality of the students reports and presentations have also been high. The course evaluation shows that the students response to the course has been very positive and that the case-centered approach has fulfilled its goals.

From the teachers perspective the case-centered approach is more demanding. The teacher must be well prepared in the case. It requires knowledge in the case to be able to control the seminars at an adequate level and to judge the students performance orderly. On the other hand, it requires less work for preparing traditional lectures, exercises, and assignments. Most of the learning, reading, and planning are conducted by the students.

References

[1] Henry Egidius, *PBL och casemetodik: hur man gör och varför*, Studentlitteratur, Lund, 1999.

[2] James A Erskine, Michiel R Leenders, Louise A Mauffette-Leenders, *Teaching With Cases*, Ivey Publishing, Ivey Business School, 2003 (3rd edition).

[3] Anna E Flynn, James D Klein, *The Influence of Discussion Groups in a Case-Based Learning Environment*, ETR&D, Vol. 49, No.3, pp.71–86, 2001.

[4] Bengt Kjellén, Konrad Lundberg, Yngve Myrman, *En handbook om Casemetodik*, Network for Case Methods Learning, The Swedish Council for the Renewal of Higher Education.

[5] Nigel Ling, David Parsons, *Problem–Based Learning as an Effective Tool for Teaching Computer Network Design*, IEEE Transactions on Education, Vol.49, No.1, pp.5–10, 2006.

[6] Ulf Melin, Stefan Cronholm, *Project Oriented Student Work – Learning & Examination*, in proceedings of ITiCSE'04, Leeds, pp.87–91, 2004.

[7] Peter R Turner, *Teaching Scientific Computing Through Projects*, in proceedings of 29:th ASEE/IEEE Frontiers in Education Conference, San Juan, Puerto Rico, 1999.